Appendix for Wasserstein Expansible Variational Autoencoder for Discriminative and Generative Continual Learning

August 18, 2023

Contents

A Additional information for the proposed WEVAE

In this section, we provide the pseudocode for the proposed methodology of the Wasserstein Expansible Variational Autoencoder (WEVAE), which also involves the testing phase, which is provided in Algorithm 1. We summarize the learning procedure of WEVAE into four steps :

- **Step 1 (Sample selection).** At a certain training time $\mathcal{T}_t$, we add a new data batch into the memory buffer, expressed as $\mathcal{M}_t = \mathcal{M}_t \bigcup \mathbf{X}_t, \mathbf{X}_t \sim \mathcal{S}$. We will perform the sample selection using the novelty criterion based on the energy function form Eq. (5) and on a threshold λ, according to Eq. (4) from the paper if the memory buffer $\mathcal{M}_t$ is not overloaded $|\mathcal{M}_t| \leq |\mathcal{M}|^{max}$.
- **Step 2 (Training process).** At the time $\mathcal{T}_t$, we train the current component $\mathcal{G}_k$ on $\mathcal{M}_t$ on a batch of samples, using Eq. (1) of the paper.
- **Step 3 (Check the model's expansion).** If the proposed WEVAE has only a single component, we then build the second component when reaching the critical mass of data samples in the buffer $\mathcal{T}_t = |\mathcal{M}|^{max}$ aiming to preserve the initial knowledge that is used for the expansion process, otherwise, we describe the expansion process as follows : If the memory buffer is full $|\mathcal{M}_t| = |\mathcal{M}|^{max}$, we check the model's expansion using Eq.(4) of the paper to reduce the computational costs. If Eq. (4) of the paper is satisfied, we add new component $\mathcal{G}_{k+1}$ into $\mathbf{G}$. We also clear up the memory buffer in order to allow the newly added component to learn non-overlapping data samples.
- **Step 4 (Testing phase).** We perform the component selection by comparing the sample log-likelihood estimated by each component and then select that component with the maximum sample log-likelihood for the evaluation.

The derivation of Eq.(7) of the paper :

The intractable marginal log-likelihood $\log p(\mathbf{x}) = \iint \log p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{y}) p(\mathbf{z}, \mathbf{y}) \, \mathrm{d}\mathbf{z} \, \mathrm{d}\mathbf{u}$

Algorithm 1 Algorithm for WEVAE

(Input:The data stream);
for $\mathcal{T}_t < \mathcal{T}_n$ **do**
  Sample selection in the memory buffer
  $\mathbf{X}_t \sim \mathcal{S}$
  $\mathcal{M}_t = \mathcal{M}_t \cup \mathbf{X}_t$
  if $|\mathcal{M}_t| > |\mathcal{M}|^{max}$ **then**
    for $t < |\mathcal{M}_t|$ **do**
      $E(\mathbf{x}'_t) = \frac{1}{k-1}\sum_{j=1}^{k-1}\left\{\mathrm{Re}(\mathbf{x}'_t, f_{\theta_j}(f_{\omega_j}(\mathbf{x}'_t))\right\}$
    end for
    $\mathcal{M}_t = \bigcup_{i=1}^{|\mathcal{M}_t|^{max}} \mathcal{M}'_t[i]$
  end if
  Training process
  if $k = 1$ and $\mathcal{T}_t = |\mathcal{M}|^{max}$ **then**
    Add the second component $\mathcal{G}_2$
  end if
  Train the current VAE component $\mathcal{G}_t$ on $\mathcal{M}_t$ using $\mathcal{L}_{ELBO}$
  Check the expansion
  if $|\mathcal{M}_i| > |\mathcal{M}|^{max}$ **then**
    if $\mathbb{E}[w_{s_1}, \cdots, w_{s_t}] > \lambda$ **then**
      Add a new Component $\mathcal{G}_{k+1}$
    end if
  end if
end for
Testing phase
for $i < n'$ **do**
  $\mathbf{x} \sim \mathcal{D}^T$
  $s^\star = \arg\max_{s=1,\cdots,k}\{\mathcal{L}_{ELBO}(\mathbf{x}; \mathcal{G}_s)\}$
  Choose $\mathcal{G}_{s^\star}$ for the evaluation.
end for

can have a lower bound according to the Jensen's inequality :

$$\begin{aligned}
\log p(\mathbf{x}) \geq & \mathbb{E}_{q(\mathbf{z},\mathbf{y}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{z},\mathbf{y})}{q(\mathbf{z},\mathbf{y}|\mathbf{x})}\right] = \mathbb{E}_{q(\mathbf{z},\mathbf{y}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{z},\mathbf{y})p(\mathbf{z})p(\mathbf{y})}{q(\mathbf{z},\mathbf{y}|\mathbf{x})}\right] \\
= & \mathbb{E}_{q(\mathbf{z}|\mathbf{x})q(\mathbf{y}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{z},\mathbf{y})p(\mathbf{z})p(\mathbf{y})}{q(\mathbf{z}|\mathbf{x})q(\mathbf{y}|\mathbf{x})}\right] \\
= & \mathbb{E}_{q(\mathbf{z},\mathbf{y}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{z},\mathbf{y})\right] \\
& + \mathbb{E}_{q(\mathbf{z}|\mathbf{x})q(\mathbf{y}|\mathbf{x})}\left[\log \frac{p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}\right] \\
& + \mathbb{E}_{q(\mathbf{z}|\mathbf{x})q(\mathbf{y}|\mathbf{x})}\left[\log \frac{p(\mathbf{y})}{q(\mathbf{y}|\mathbf{x})}\right]
\end{aligned} \tag{1}$$

where we also consider the independence between the variables $\mathbf{y}$ and $\mathbf{z}$. Then according to the KL divergence form, Eq. (1) can be rewritten as :

$$\begin{aligned}\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z},\mathbf{y}|\mathbf{x})}\left[\log p(\mathbf{x} \mid \mathbf{z},\mathbf{y})\right] \\ - D_{KL}(q(\mathbf{z}\,|\,\mathbf{x}) \,||\, p(\mathbf{z})) \\ - D_{KL}(q(\mathbf{y}\,|\,\mathbf{x}) \,||\, p(\mathbf{y}))\,.\end{aligned} \tag{2}$$

where we omit the subscripts from Eq. (1) of the paper, for the sake of simplification.

B The proof of Theorem 1

In this section, we provide the detailed proof according to the results from [13]. First, we consider a single component of WEVAE $\mathcal{G}_c^i$, which is trained on $\mathcal{M}_i$ at $\mathcal{T}_i$. We have the following equation according to [13] :

$$\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq -\mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i},\mathbb{P}_{\widetilde{\mathbf{x}}_c^i}) - \frac{1}{2}\log\pi\,, \tag{3}$$

We then add $-\mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i})$ in both sides of Eq. (3), resulting in :

$$\begin{aligned}&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] - \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{G_c^i}) \leq -\mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i}) \\ &- \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}},\mathbb{P}_{\mathcal{G}_c^i}) \\ &- \frac{1}{2}\log\pi\,,\end{aligned} \tag{4}$$

where $\mathrm{W}^\star_{\mathcal{L}}(\cdot,\cdot)$ is defined in Eq. (12) from the paper.

The first term in the right-hand side (RHS) of Eq. (4) is bounded according to [13] :

$$\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathcal{G}_c^i(\mathbf{z}))] \leq -\mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i}). \tag{5}$$

From Eq. (5), we have :

$$\begin{aligned}&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathcal{G}_c^i(\mathbf{z}))] \\ &+ \Big| \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}\mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathcal{G}_c^i(\mathbf{z}))] - \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i})\Big| \geq \\ &- \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i}),\end{aligned} \tag{6}$$

We then replace the first term in the RHS of Eq. (4) by the above equation, resulting in :

$$
\begin{aligned}
&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] - \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i}) \\
&\leq \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathcal{G}_c^i(\mathbf{z}))] - \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}},\mathbb{P}_{\mathcal{G}_c^i}) \\
&+ \Big| \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathcal{G}_c^i(\mathbf{z}))] - \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i}) \Big| \\
&- \frac{1}{2}\log\pi\,,
\end{aligned}
\tag{7}
$$

We then add the negative KL divergence term in both sides of Eq. (7) :

$$
\begin{aligned}
&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] - \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i}) \\
&- \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z}))] \leq \\
&\underbrace{\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathcal{G}_c^i(\mathbf{z}))] - \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z}))] - \frac{1}{2}\log\pi}_{\text{ELBO}} \\
&- \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}},\mathbb{P}_{\mathcal{G}_c^i}) + \Big| \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathcal{G}_c^i(\mathbf{z}))] - \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i}) \Big|\,,
\end{aligned}
\tag{8}
$$

According to the definition of ELBO, Eq. (8) can be rewritten as :

$$
\begin{aligned}
&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \\
&- \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i}) - \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z}))] \leq \\
&\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] - \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}},\mathbb{P}_{\mathbf{G}_i}) \\
&+ \Big| \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathcal{G}_c^i(\mathbf{z}))] - \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i}) \Big|\,,
\end{aligned}
\tag{9}
$$

Then we rewrite Eq. (9), resulting in :

$$
\begin{aligned}
\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq{}& \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] + \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i}) \\
&- \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}},\mathbb{P}_{\mathcal{G}_c^i}) \\
&+ \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z}))] \\
&+ \Big| \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x},\mathcal{G}_c^i(\mathbf{z}))] \\
&- \mathrm{W}^\star_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i},\mathbb{P}_{\mathcal{G}_c^i}) \Big|\,,
\end{aligned}
\tag{10}
$$

We consider that $\mathcal{L}(\cdot)$ satisfies triangle inequality, we have :

$$\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_{\mathcal{G}_c^i}) + \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathbf{x}}, \mathbb{P}_{\mathcal{G}_c^i}) \geq \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\mathcal{M}_i}) \tag{11}$$

We move the second term in the left-hand side of Eq. (11) in the right-hand side :

$$\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\mathcal{G}_c^i}) \geq \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\mathcal{M}_i}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_{\mathcal{G}_c^i}) \tag{12}$$

Then we replace $\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\mathcal{G}_c^i})$ from Eq. (10) by the expression of Eq. (12), resulting in :

$$\begin{aligned}\inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq & \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \\ & + 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_{\mathcal{G}_c^i}) \\ & - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\mathcal{M}_i}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathcal{G}_c^i}, \mathbb{P}_{\mathcal{M}_i}),\end{aligned} \tag{13}$$

where $\tilde{\mathrm{F}}(\mathbb{P}_{\mathcal{G}_c^i}, \mathbb{P}_{\mathcal{M}_i})$ is expressed as :

$$\begin{aligned}\tilde{\mathrm{F}}(\mathbb{P}_{\mathcal{G}_c^i}, \mathbb{P}_{\mathcal{M}_i}) = & \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[D_{KL}(q_\omega(\mathbf{z}\,|\,\mathbf{x})\,||\,p(\mathbf{z}))] \\ & + \left| \inf_{q_\omega(\mathbf{z})=p(\mathbf{z})} \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}} \mathbb{E}_{q_\omega(\mathbf{z}\,|\,\mathbf{x})}[-\mathcal{L}(\mathbf{x}, \mathcal{G}_c^i(\mathbf{z}))] - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_{\mathcal{G}_c^i}) \right|\end{aligned} \tag{14}$$

For the sake of simplification we omit $\inf$ in Eq. (13), resulting in :

$$\begin{aligned}\mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \leq & \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\theta,\omega)] \\ & + 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_{\mathcal{G}_c^i}) \\ & - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\mathcal{M}_i}) + \tilde{\mathrm{F}}(\mathbb{P}_{\mathcal{G}_c^i}, \mathbb{P}_{\mathcal{M}_i}),\end{aligned} \tag{15}$$

This results in the derivation of a bound for a single component $\mathcal{Q}_c^i$. We can easily extend Eq. (15) for a WEVAE mixture model $\mathbf{G} = \{\mathcal{G}_1, \cdots, \mathcal{G}_k\}$, resulting in :

$$\begin{aligned}&\textstyle\sum_{j=1}^{a_i} \sum_{t=1}^{c_j} \left\{ \left\{ \widetilde{\mathrm{F}}_s(\mathbf{G}, \mathbb{P}_{\mathbf{x}_i^j(t)}) \right\} \right\} \leq \\ &\textstyle\sum_{j=1}^{a_i} \sum_{t=1}^{c_j} \left\{ \left\{ \mathrm{F}_s(\mathbf{G}, \mathbb{P}_{\mathbf{x}_i^j(t)}) \right\} \right\} \\ &\square\end{aligned} \tag{16}$$

which corresponds to Eq. (9) from the paper.

C The proof of Theorem 2

Let us consider a WEVAE model $\mathbf{G}$ with k components, we can view $\mathbf{G}$ as a single model trained on all memories $\{\mathcal{M}_{b_1}, \cdots, \mathcal{M}_{b_k}\}$, based on the component selection (Eq.(10) of the paper). Let $\mathbb{P}_{\widetilde{\mathbf{x}}^i}$ be the distribution of samples equaly drawn from each component $\{\mathcal{G}_j, j = 1, \cdots, |\mathbf{G}|\}$ at $\mathcal{T}_i$. Let $\mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}$ be the distribution of all memories $\{\mathcal{M}_{b_1}, \cdots, \mathcal{M}_{b_k}\}$. Based on Eq. (15), we have :

$$\begin{aligned}
&\mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}_i}}[\mathcal{L}_{ELBO}(\mathbf{x}; \mathbf{G})] \leq \mathbb{E}_{\mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}}[\mathcal{L}_{ELBO}(\mathbf{x}; \mathbf{G})] \\
&+ 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}, \mathbb{P}_{\widetilde{\mathbf{x}}^i}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}) \\
&+ \widetilde{\mathrm{F}}(\mathbb{P}_{\widetilde{\mathbf{x}}^i}, \mathbb{P}_{\mathcal{M}_{b_{1:b_k}}}),
\end{aligned} \tag{17}$$

$\square$

In the following, we compare the proposed WEVAE model with the static model under the theoretical framework defined in the paper. We start with providing the definition of the static model.

Definition 4. Let $\mathcal{G}^i$ be a single/static model which is trained on $\mathcal{M}_i$ at $\mathcal{T}_i$. Let $\mathbb{P}_{\mathcal{G}^i}$ be the distribution of samples drawn from the generation process of $\mathcal{G}^i$ at $\mathcal{T}_i$.

Lemma 1. Based on **Definition 4.**, we can derive a bound for a single/static model at $\mathcal{T}_i$:

$$\begin{aligned}
&\mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}_i}}[\mathcal{L}_{ELBO}(\mathbf{x}; \mathcal{G}^i)] \leq \mathbb{E}_{\mathbb{P}_{\mathcal{M}_i}}[\mathcal{L}_{ELBO}(\mathbf{x}; \mathcal{G}^i)] \\
&+ 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_i}, \mathbb{P}_{\mathcal{G}^i}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\mathcal{M}_i}) \\
&+ \widetilde{\mathrm{F}}(\mathbb{P}_{\mathcal{G}^i}, \mathbb{P}_{\mathcal{M}_i}),
\end{aligned} \tag{18}$$

Eq. (18) when compared to Eq. (17), has smaller upper bound since the term $\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\mathcal{M}_{b_{1:b_k}}})$ in the RHS of Eq. (17) can be reduced significantly when training more components. This shows that the proposed WEVAE naturally performs better than the single/static model.

D Analysis for selecting λ

In this section, we theoretically analyze the role of the threshold λ used for model expansion in Eq. (4) and the model's generalization performance. According to Theorem

2, we have :

$$\begin{aligned} \mathbb{E}_{\mathbb{P}_{\widehat{\mathbf{x}}_i}}[\mathcal{L}_{ELBO}(\mathbf{x};\mathbf{G})] &\le \mathbb{E}_{\mathbb{P}_{\mathcal{M}_{b_1:b_k}}}[\mathcal{L}_{ELBO}(\mathbf{x};\mathbf{G})] \\ &+ 2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_{b_1:b_k}}, \mathbb{P}_{\widetilde{\mathbf{x}}^i}) - \mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\widehat{\mathbf{x}}_i}, \mathbb{P}_{\mathcal{M}_{b_1:b_k}}) \\ &+ \widetilde{\mathrm{F}}(\mathbb{P}_{\widetilde{\mathbf{x}}^i}, \mathbb{P}_{\mathcal{M}_{b_1:b_k}})\,. \end{aligned} \tag{19}$$

A large threshold λ encourages the model to frequently build new components, resulting in a model with many components (k in Eq. (19) is large). Then the distribution $\mathbb{P}_{\mathcal{M}_{b_1:b_k}}$ would preserve more knowledge from the data stream and can thus reduce the term $2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_{b_1:b_k}}, \mathbb{P}_{\widetilde{\mathbf{x}}^i})$ in Eq. (19), leading to better performance. In contrast, if we consider a small λ, this would prevent WEVAE model's expansion, leading to fewer components. Therefore, the distribution $\mathbb{P}_{\mathcal{M}_{b_1:b_k}}$ would miss some underlying data distributions from the data stream and the term $2\mathrm{W}^{\star}_{\mathcal{L}}(\mathbb{P}_{\mathcal{M}_{b_1:b_k}}, \mathbb{P}_{\widetilde{\mathbf{x}}^i})$ is increased in Eq. (19), leading to worse performance. The optimal threshold λ should ensure a good trade-off between the model size and its resulting generalization performance. Optimally, this would be implemented by ensuring that each individual WEVAE component models a unique data distribution. In this way we would minimize the overlap between the statistical representations by two different components and WEVAE would represent a diversity of distributions while using an optimal number of parameters.

E Additional information for the experiment setting

The release of the code. We have provided the detailed implementation of the proposed Wasserstein Expansible Variational Autoencoder (WEVAE) model. We also provide the source code in the supplemental material. In addition, We will provide after properly organizing the source code used in the experiments and for the testing of the WEVAE model for the sake of easy understanding and for facilitating the re-implementation and we will release it publicly on https://github.com/, if the paper is accepted.

E.1 Experiment setting

The hyperparameter configuration and GPU hardware. For all experiments, we use Adam [6] with a learning rate of 0.0001 and its default hyperparameters. For the density estimation task, we employ the batch size of 64 and one training epoch for training. All experiments are performed on the server with the operating system Ubuntu 18.04.5. We also use the GPU (NVIDIA A40) for all our experiments.

The configuration of the network architecture for density estimation task. Following from [3], we use two fully connected layers for implementing the generator and inference models. Each layer in the neural network has 200 hidden units. The maximum memory size for Split MNIST, Split Fashion, Split MNIST-Fashion, Cross-domain is 1.5K, 1.5K, 1.9K and 2.0K, respectively.

Additional information for the evaluation. All results reported in the paper are evaluated on the testing datasets after the task-free continual learning.

E.2 The configuration for the classification task.

In this section, we provide the detailed information for the classification task. First, we employ several datasets inlcuding Split MNIST, Split CIFAR10, Split CIFAR100 and Split MiniImageNet, which are introduced in the following.

Split MNIST. We divide MNIST which contains 60k training samples into five tasks, each consisting of images from two classes, in consecutive order of their displayed digits, while increasing the numbers represented in the images [4].

Split CIFAR10. We split CIFAR10 into five tasks where each task consists of samples from two different classes [4].

Split CIFAR100. We split CIFAR100 into 20 tasks where each task has 2500 examples from five different classes [8].

Split MiniImageNet. We divide the MiniImageNet into 20 tasks [10], where each task collects the images of five classes [2].

In the following, we describe the detailed information of the network architecture used in our classification task.

We adapt ResNet 18 [5] for Split CIFAR10 and Split CIFAR100. We use an MLP network with 2 hidden layers of 400 units each [4] for Split MNIST. The maximimum memory size for Split MNIST, Split CIFAR10, Split CIFAR100 are 2000, 1000 and 5000, respectively. At the testing phase, we make the component selection by comparing the sample log-likelihood and the classifier of the selected component is used for prediction.

We introduce additional information for several baselines, used in the experimental results from the Tables 1-4 from the paper, in the following.

Finetune trains a single model directly on a new batch of images during the online continual learning.

Gradient Episodic Memory (GEM) [8] is a memory-based approach that would use the memory to store past samples. GEM is also required to access both the task label and class label during the training.

Dynamic-OCM [13] is a dynamic expansion model which proposes an online cooperative memorization (OCM) approach. OCM manages two memory buffers, aiming to store short- and long-term knowledge during training. In addition, Dynamic-OCM detects the change of the loss value as expansion signals, which does not have theoretical guarantees.

Incremental Classifier and Representation Learning (iCARL) [9] is a standard memory-based method used in a class incremental setup.

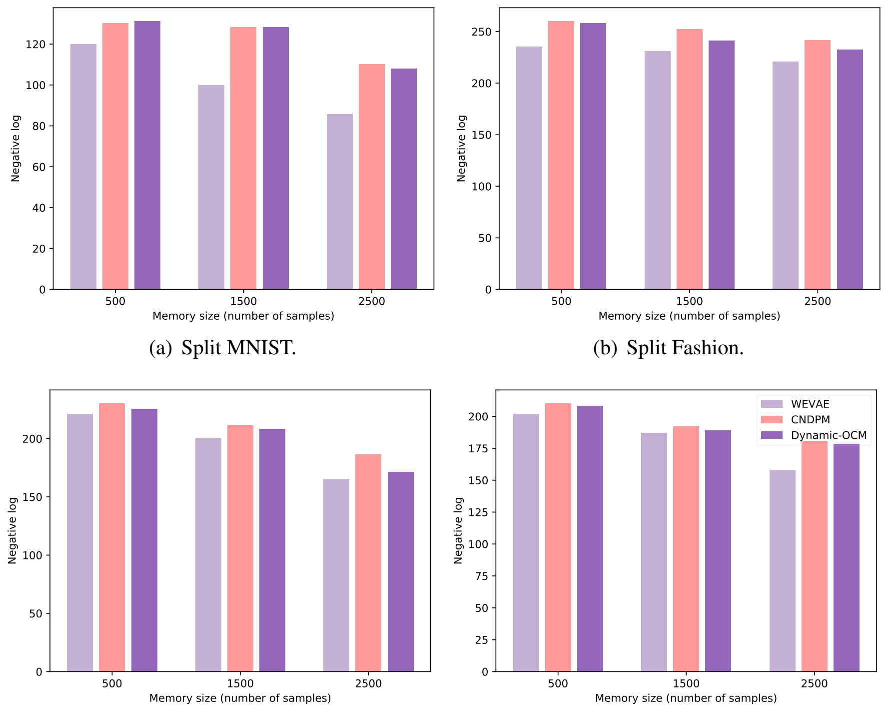
reservoir* [11] is a memory-based approach that stores the observed samples into a memory buffer $\mathcal{M}$ with probability $|\mathcal{M}|/n$ where n is the number of stored samples, and $|\cdot|$ represents the cardinality of a set.

MIR [2] introduces a retrieval strategy for the sample selection in the memory during the Online Continual Learning (OCL). However, the retrieval strategy in MIR requires evaluating the loss in each training session. This means that MIR requires modifying the retrieval strategy for different tasks such as classification or generation tasks.

GSS [1] formulates the sample selection process as a constraint reduction problem. GSS stores samples in a buffer using the gradient information which requires to access the class labels and can not be applied in the unsupervised learning setting.

E.3 The configuration for the density estimation task

Following from [13], we compare our model (WEVAE) with existing TFCL methods in the density estimation task, which are outlined as : (1) VAE-ELBO-OCM : A single VAE model with ELBO using the Online Cooperative Memorization (OCM) [13]. (2) VAE-IWVAE50-OCM : A single VAE model with IWVAE using the OCM where the number of importance samples is 50. (3) VAE-ELBO-Random : A single VAE model with a memory that randomly removes samples when it reaches the maximum memory size. (4) Dynamic-ELBO-OCM : A mixture model with ELBO using OCM [13]. (5) CNDPM [7]; (6) LIMix [12] : We assign an episodic memory with a fixed buffer size for the LIMix model used for TFCL. The maximum number of components for various models is set to 30 to avoid memory overload. For the classification task, we adopt the baselines from the recently adopted TFCL benchmark from [4].



(a) Split MNIST.

(b) Split Fashion.

(c) Split MNIST-Fashion.

(d) Cross domain.

Figure 1: The performance of various models on four datasets when changing the memory size.

F Additional results for ablation study

In this section, we provide additional results for the ablation study, which investigate the effectiveness of each module of the proposed WEVAE.

F.1 Changing the memory size

We evaluate the performance of various models by changing the memory size, and the results are provided in Fig. 1. As the memory buffer increases its capacity, all models improve their performance. The proposed WEVAE outperforms other models on all memory configurations, even if the memory buffer can only store 500 samples. In addition, the proposed WEVAE outperforms other baselines by a large margin when having enough memorized samples.

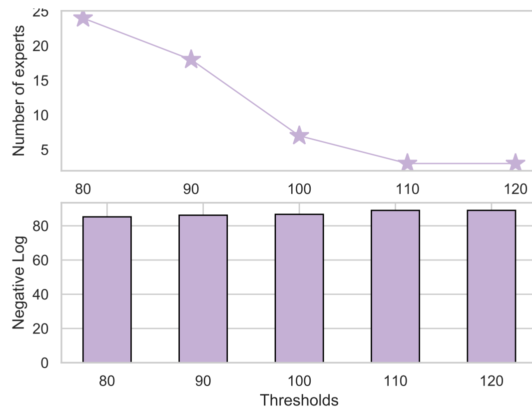


Figure 2: The performance and the number of components of WEVAE when changing the expansion threshold λ in the density estimation task on Split MNIST.

F.2 Changing the threshold λ

We investigate the performance and the number of components of WEVAE on Split MNIST when changing the threshold λ and the results are shown in Fig. 2. Decreasing λ can increase the number of components but does not lead to a significant improvement in the performance. These results show that the proposed WEVAE can achieve good performance using only three components, demonstrating that each component in WEVAE can capture different knowledge well.

F.3 Changing the batch size

We also investigate the performance and the number of components of WEVAE when changing the batch size, and the results are shown in Fig. 3 on Split MNIST dataset. We can observe that the proposed WEVAE does not suffer from a degenerated performance and maintains a similar number of components when changing the batch size.

F.4 Model expansion process

We investigate the number of components of WEVAE and the change of the distribution (task) on Split MNIST in the classification task, and the results are reported in Fig. 4. The proposed WEVAE frequently creates components at the initial learning stage instead of the later learning stages. The reason is that when WEVAE has accumulated more knowledge, it does not need more components to learn the related information

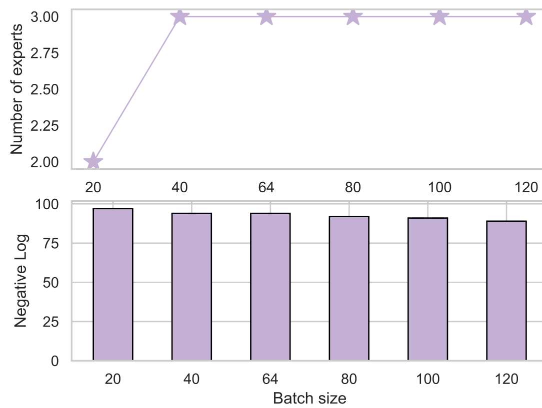


Figure 3: The performance and the number of components of WEVAE when changing the batch size on Split MNIST.

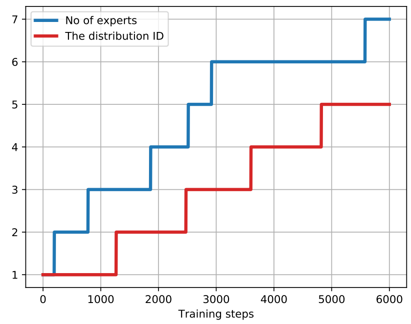
in the later learning process. In addition, a small threshold λ encourages WEVAE to build more components. A suitable threshold λ such as 60 enables the WEVAE to employ a reasonable number of components where each component captures a unique underlying data distribution.

F.5 Fuzzy task setting

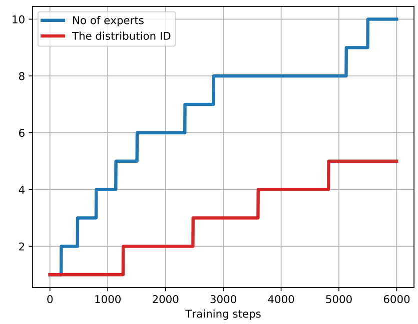
In a realistic continual learning setting, a model usually accesses samples drawn from a data stream with fuzzy task boundaries [7]. In this section, we evaluate the performance of various models on the fuzzy task setting. We employ the same procedure as in [7], which swaps randomly chosen samples between two tasks from each data stream. The results are reported in Tab. 1, which show that the proposed WEVAE outperforms other baselines on the fuzzy task setting.

F.6 Comparison with another sample selection approach

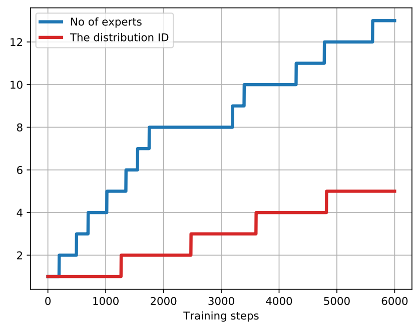
We create two baselines WEVAE-GSS and WEVAE reservoir, which employ GSS and Reservoir, respectively, for sample selection. The results for the classification task are provided in Tab. 2. These results demonstrate that the proposed sample selection approach outperforms using the Reservoir's sample selection approach in all datasets considered.



(a) $\lambda = 60$.



(b) $\lambda = 55$.



(c) $\lambda = 50$.

Figure 4: The number of components of WEVAE and the change of the distribution on Split MNIST in the classification task.

F.7 Computational complexity analysis

We investigate the computational costs of the proposed WEVAE in the classification task, and the results are provided in Tab. 3. We find that the proposed WEVAE requires more training time than CNDPM. This is because the proposed dynamic expansion mechanism uses the generative replay process, leading to additional computational costs. However, the proposed WEVAE outperforms CNDPM by a large margin on various tasks while requirying similar computational times compared with CNDPM. Furthermore, to compare with Dynamic-OCM, the proposed WEVAE is more efficient since Dynamic-OCM requires performing the sample selection for the memory buffer. As a result, the proposed WEVAE outperforms Dynamic-OCM on both the density estimation and classification tasks.

Methods	Split MNIST	Split CIFAR10	Split MImageNet
Vanilla	21.53 $\pm$ 0.1	20.69 $\pm$ 2.4	3.05 $\pm$ 0.6
ER	79.74 $\pm$ 4.0	37.15 $\pm$ 1.6	26.47 $\pm$ 2.3
MIR	84.80 $\pm$ 1.9	38.70 $\pm$ 1.7	25.83 $\pm$ 1.5
ER + GMED	82.73 $\pm$ 2.6	40.57 $\pm$ 1.7	28.20 $\pm$ 0.6
MIR+GMED	86.17 $\pm$ 1.7	41.22 $\pm$ 1.1	26.86 $\pm$ 0.7
WEVAE	**88.78** $\pm$ 1.2	**45.26** $\pm$ 1.8	**30.12** $\pm$ 1.2
WEVAE-NoS	87.65 $\pm$ 1.3	44.97 $\pm$ 1.3	29.57 $\pm$ 0.9

Table 1: The classification accuracy of five independent runs for various models over data streams with fuzzy task boundaries.

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
WEVAE-GSS	94.32	51.98	22.62
WEVAE-reservoir	94.12	51.02	22.18
WEVAE	**96.63**	**54.98**	**25.03**

Table 2: The classification accuracy of various models on three datasets, respectively.

F.8 The knowledge diversity among WEVAE's components

We investigate whether the proposed WEVAE can train its mixture components to learn diverse information during the training. We train WEVAE on Split MNIST in the classification task. After training, the proposed WEVAE builds seven components and we show the results for the data generated by each component in Fig. 5. We can observe that each component generates images belonging to a different underlying data distribution, demonstrating that the proposed WEVAE can train its components, each being characterized by a different probabilistic representtaion, which is consistent with our theoretical analysis from **Theorem 2** of the paper.

F.9 Comparison to the task-aware baselines

In this section, we compare the proposed WEVAE with the task-aware approaches on a long sequence of tasks. According to the setting from [12], we consider a sequence

Methods	Split MNIST	Split CIFAR10	Split CIFAR100
WEVAE	1.3	20.02	33.52
CNDPM	0.9	18.6	30.23
Dynamic-OCM	10.2	42.3	47.8

Table 3: The training time (minutes) of various models.

(a) Generations of component 1.

(b) Generations of component 2.

(c) Generations of component 3.

(d) Generations of component 4.

(e) Generations of component 5.

(f) Generations of component 6.

(g) Generations of component 7.

Figure 5: The generation of each expert in DSVitE on Split MNIST

of several databases, including MNIST, Fashion, SVHN, Inverse Fashion (IFashion), Rotate MNIST (RMNIST), resulting in the sequence MSFIR. We assign a memory buffer that can store maximum 5000 samples for the proposed WEVAE. The batch size is 64 and the results are reported in Tab. 4 where the results of all comparison baselines are taken from [12]. These results show that the proposed WEVAE still performs other methods even if the task information is not provided.

F.10 Analysis for the model complexity

In this section, we analyze the model complexity of various models under the density estimation task. The number of parameters of various models are reported in Tab. 5.

	MSE					
Datasets	LGM	CURL	BE	GMM	Stud	WEVAE
MNIST	129.93	211.21	19.24	26.64	176.82	67.41
Fashion	89.28	110.60	38.81	33.67	178.04	92.56
SVHN	169.55	102.06	39.57	30.27	146.70	114.63
IFashion	432.90	115.29	36.52	35.03	158.18	59.09
RMNIST	130.28	279.47	25.41	22.97	157.55	68.68
Average	190.38	163.72	31.91	**29.71**	163.45	80.47

Table 4: The performance of various models after MSFIR lifelong learning.

Methods	**Split MNIST**	**Split Fashion**	**Split MNIST-Fashion**	**Cross domain**
WEVAE	6M	20M	16M	18M
WEVAE-NoS	10M	20M	16M	18M
LIMix	60M	60M	60M	60M
CNDPM	60M	60M	60M	60M
Dynamic-ELBO-OCM	10M	20M	20M	22M

Table 5: The number of parameters of various models under the density estimation task. 'M' represents millions of parameters. WEVAE-NoS, represents the situation where we do not consider the sample selection mechanism, as described in Section 4.2 in the paper.

These results show that the proposed WEVAE employs equal or fewer parameters while achieving better performance than other dynamic expansion models.

F.11 The effect when not considering the stochastic process

In this section, we investigate the effect of the proposed WEVAE without using the stochastic process. Eq.(3) of the paper can be rewritten as the expansion criterion :

$$\min \left\{ (\mathcal{L}_d(\mathbb{P}_{\theta_1^{t_1}}, \mathbb{P}_{\theta_k^t}), \cdots, \mathcal{L}_d(\mathbb{P}_{\theta_{k-1}^{t_{k-1}}}, \mathbb{P}_{\theta_k^t}) \right\} \geq \lambda \,, \tag{20}$$

We call WEVAE using Eq. (20) as WEVAE-1. We train both WEVAE and WEVAE-1 using the same hyperparameter configuration on Split MNIST, Split CIFAR10 and

Methods	Split MNIST	N	Split CIFAR10	N	Split CIFAR100	N
WEVAE	96.87	5	55.26	6	25.12	5
WEVAE-1	95.75	7	54.12	8	24.74	6

Table 6: Classification accuracy of various models on three datasets.

Split CIFAR100. The classification results are reported in Tab. 6, which show that WEVAE outperforms WEVAE-1 while employing fewer components. These results demonstrate that the stochastic process can further improve the performance and reduce the number of parameters for WEVAE.

References

[1] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11817–11826, 2019. 11

[2] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11872–11883, 2019. 10, 11

[3] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015. 10

[4] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8250–8259, 2021. 10, 11

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recog. (CVPR)*, pages 770–778, 2016. 10

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1412.6980*, 2015. 9

[7] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural Dirichlet process mixture model for task-free continual learning. In *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2001.00689*, 2020. 11, 14

[8] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6467–6476, 2017. 10

[9] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010, 2017. 11

[10] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *Advances in neural information processing systems (NIPS)*, 29:3637–3645, 2016. 10

[11] Jeffrey Vitter. Random sampling with a reservoir. *ACM Trans. on Mathematical Software (TOMS)*, 11(1):37–57, 1985. 11

[12] Fei Ye and Adrian G. Bors. Lifelong infinite mixture model based on knowledge-driven Dirichlet process. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pages 10695–10704, 2021. 11, 16, 17

[13] Fei Ye and Adrian G. Bors. Continual variational autoencoder learning via online cooperative memorization. In *Proc. European Conference on Computer Vision (ECCV), vol. LNCS 13683*, pages 531–549, 2022. 5, 11